Experimentation in the Use of Service Orientation in Resource-Constrained Environments

May 18, 2011

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Joe Seibel
Dan Plakosh
Soumya Simanta
Ed Morris
William Anderson

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **18 MAY 2011**	2. REPORT TYPE	3. DATES COVERED
4. TITLE AND SUBTITLE **Experimentation in the Use of Service Orientation in Resource-Constrained Environments**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Carnegie Mellon University ,Software Engineering Institute,Pittsburgh,PA,15213**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**27**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Scope

Current & Future Reality

- "Last Mile" users in tactical military situations & early responders in crisis situations desire
  - mobile, dismounted, interoperating coalitions, using ad hoc and wireless networks, to improve situational capabilities

Purpose & Goals of the IRAD

- Determine if SOA is applicable and implementable in tactical environments

Success Criteria

- Sound understanding, environment characterization, functioning prototypes, and documented lessons

Related Work

Apps for the Army

- Competition to encourage app development (non-tactical)

MITRE

- Counter-Insurgency Intelligence Collection

U.S. Department of Homeland Security and NASA

- Chemical detection using sensors on smartphone

General Dynamics

- Militarized wearable computer (GD300) based on Android

Raytheon

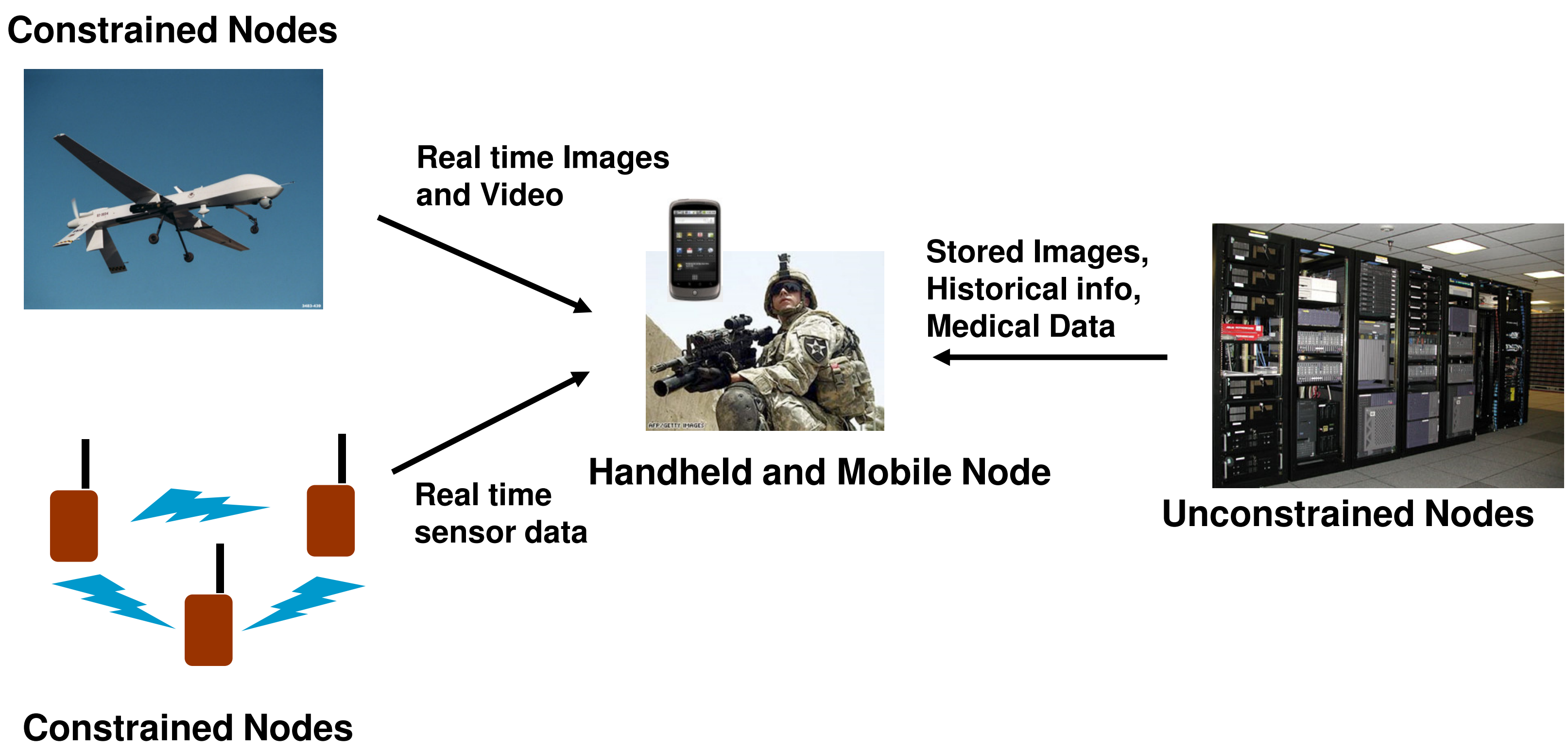
- Modification of Android for military use

Related Work

DARPA

- Transformative Apps program (DARPA-BAA-10-41)
  - Marketplace, infrastructure and apps
  - App Issues:
    - frequent disconnection, limited bandwidth
    - distributed compute/storage nodes in vehicles or outposts.
    - security tradeoffs with usability, performance, and complexity.

Feasibility Study: Communication Architecture Using Web Services



Arrows represent communication via web services

Engineering Decision: Transport Layer Protocol

Transport layer protocol defines interfaces available to applications that allow end-to-end communications; TCP is the most familiar

- TCP provides reliable transmission – data guaranteed to reach destination in correct order and without duplication.
- TCP is suited for situations with reliable transmission (solid network infrastructure)
- Not suited to situations where packet loss, mis-ordering, or garbling are more common

We selected UDP as the preferred protocol for:

- better for time-sensitive applications
- *but* UDP does not provide error correction

Engineering Decision: Message Protocol

SOA uses two common messaging protocols

- Traditional WS-* based web services employing SOAP messages
- Representational State Transfer (REST) web services using URI's, and HTTP operations

REST is simpler and increasingly more common

- Better support on the Android platform
- Close tie to HTTP and therefore TCP meant we could not use it

We selected SOAP also hoping to take advantage of well-defined specifications, open source implementations, and support for security.

Engineering Decision: SOAP Engine

SOAP is a platform/language independent XML-based protocol for encoding messages

- Variety of open source SOAP engines that serialize in-memory data structures to XML messages and vice versa
- Machines at the other end of the wire perform inverse operation

SOAP engines selected

- gSOAP on the CoT router-side supports SOAP over UDP and C++ development
- We modified kSOAP on the Android side to support UDP, but problems remain
  - We had to manually develop code to marshal/unmarshal Java objects to SOAP messages
  - kSOAP for Android omits important features (XML attributes, limited parsing, WS-Addressing)

Reduced bandwidth on GPRS networks later necessitated a change to a binary format

Engineering Decision: Security

Typical strategy for web services over the internet is

- Network level security (encryption) based on IPsec
- Transport level security built on transport mechanisms such as HTTP
- Application level security strategies such as WS-Security

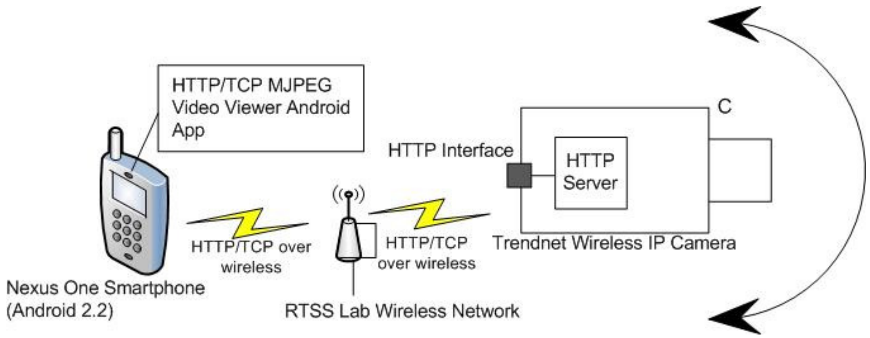
We considered WS-Security, but rejected it due to lack of kSOAP support

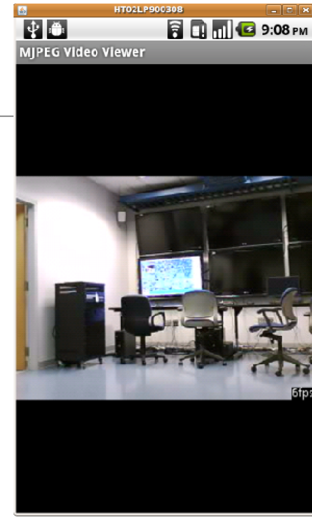
Our approach

- Network level security: existing internet and wireless protocol WPA/WPA2 with pre-shared key (PSK)
- Transport level security: Datagram Transport Layer Security (DTLS) was immature when we evaluated it
- Application level security: AES-256 bit encryption with WPA2 passphrase generation

Not perfect, but this solution provides a reasonable level of security

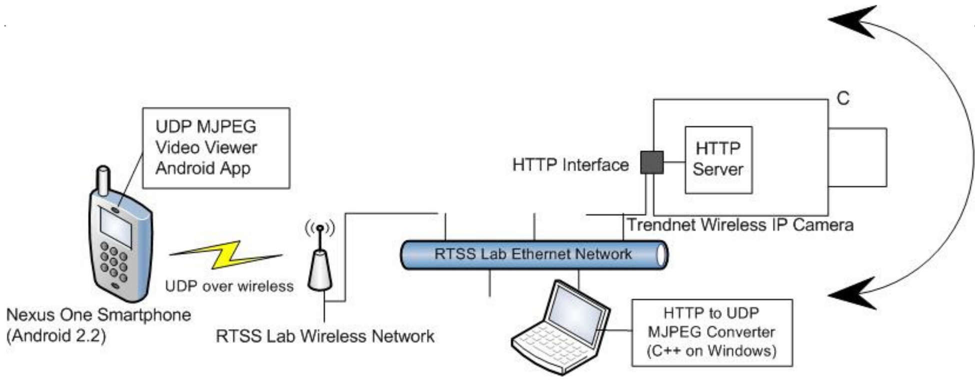
Test 1 – HTTP/TCP MJPEG Video Viewer App

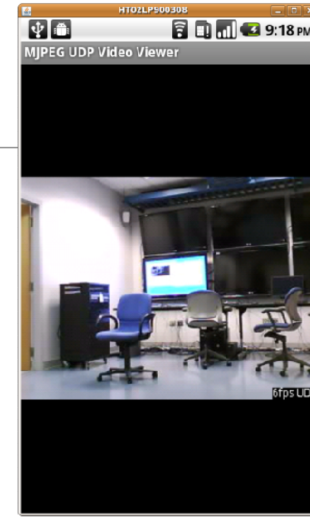




- Trendnet wireless IP Camera and Nexus One smartphone are connected to RTSS wireless network
- The smartphone receives video in Motion JPEG (MJPEG) format over a wireless HTTP/TCP connection

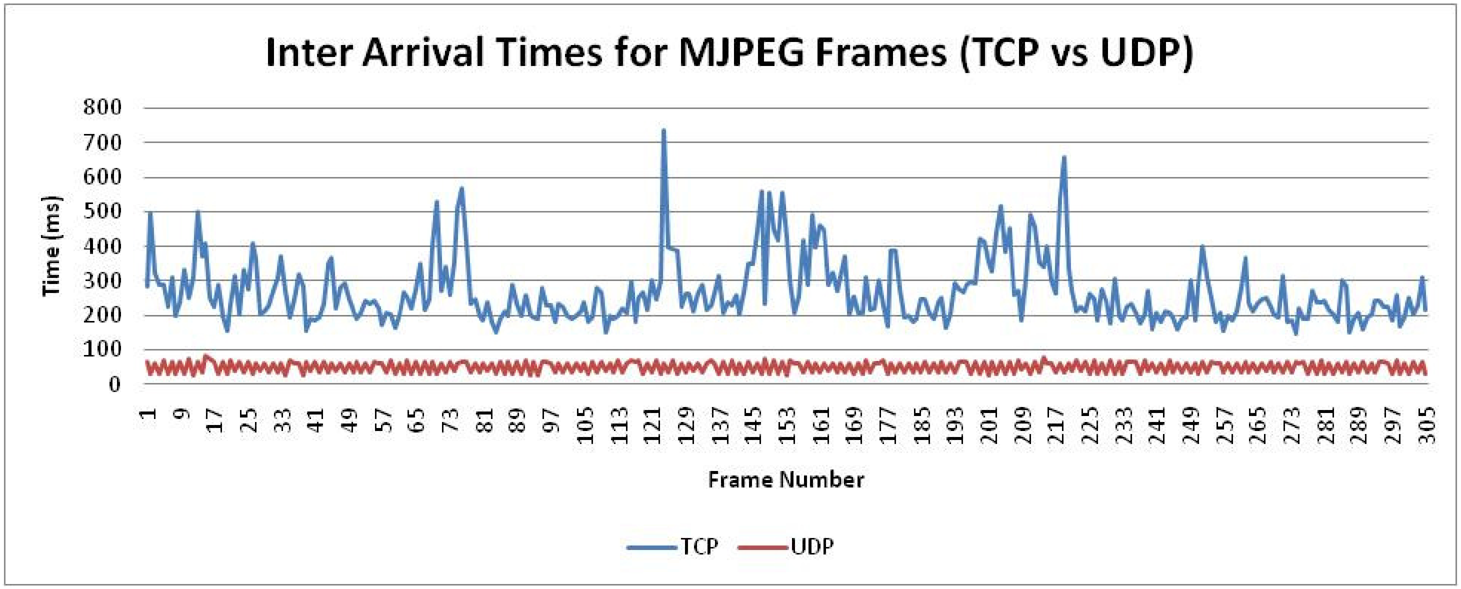
Test 2 – UDP MJPEG Video Viewer App



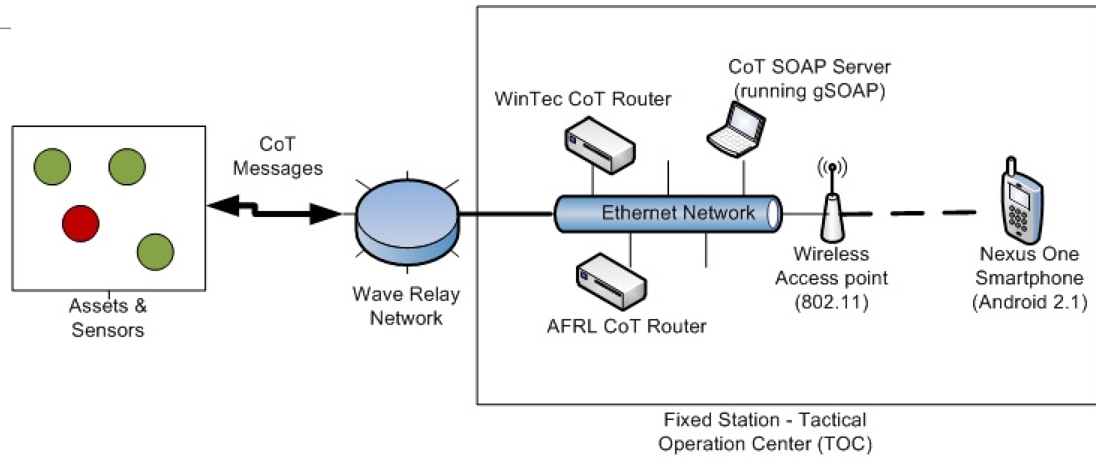


- Trendnet wireless IP Camera is connected to a Windows machine running a TCP to UDP MJPEG converter (C++) through a wired Ethernet network
- The TCP to UDP converter gets MJPEG image data over TCP and retransmits that same data using UDP to the RTSS Lab wireless network.
- The Android smartphone is connected to the RTSS lab wireless network and can receive these MJPEG frames over UDP and display it using an Android application

Performance of TCP vs UDP for Video Data on the Phone

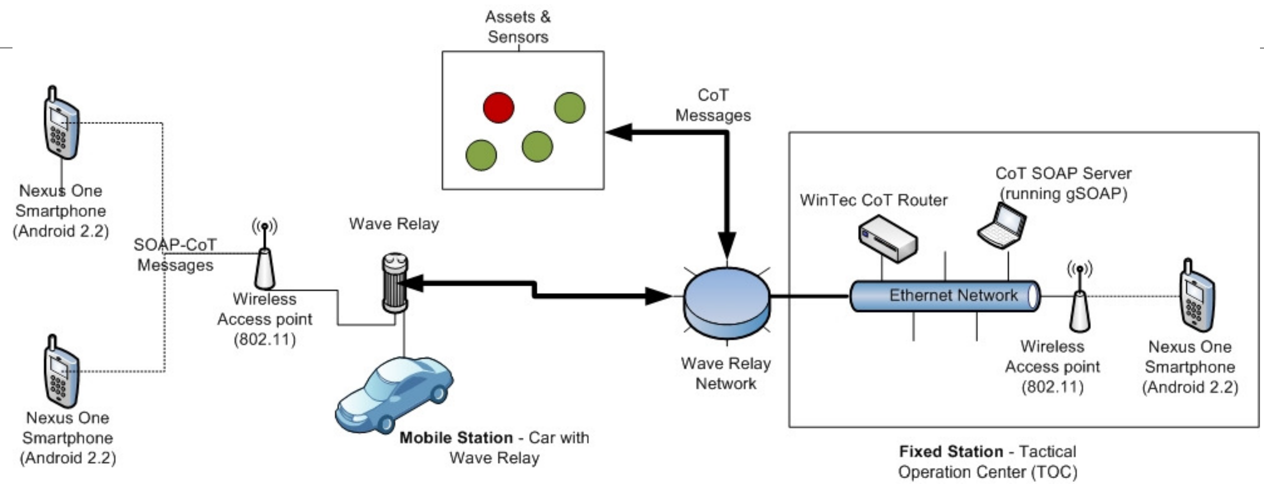


Prototype 1 – CoT SOAP UDP App V1 (Camp Roberts – May 2010 TNT) – Fixed Station

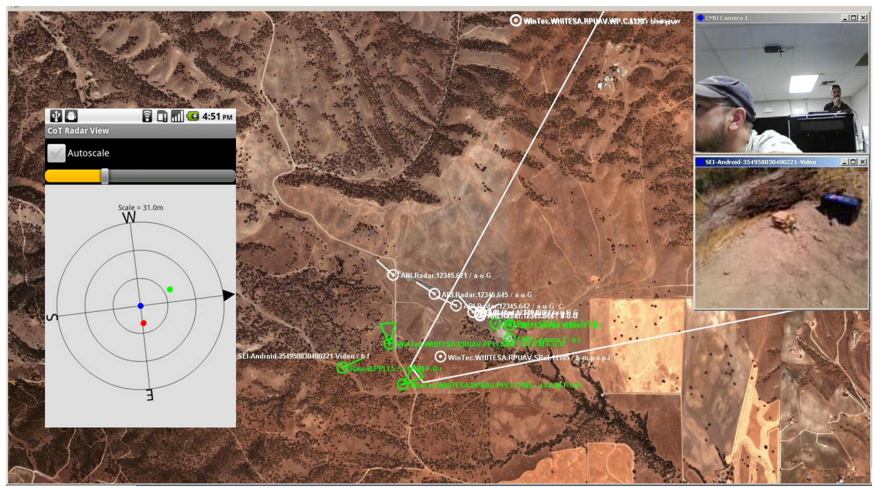


- Assets (UAVs, cars) track a hostile vehicle and post CoT messages (video, location etc) to the CoT SOAP Server
- CoT SOAP Server consume raw CoT messages and provides CoT data as SOAP-over-UDP web service
- Android phone consume SOAP messages, processes and displays them

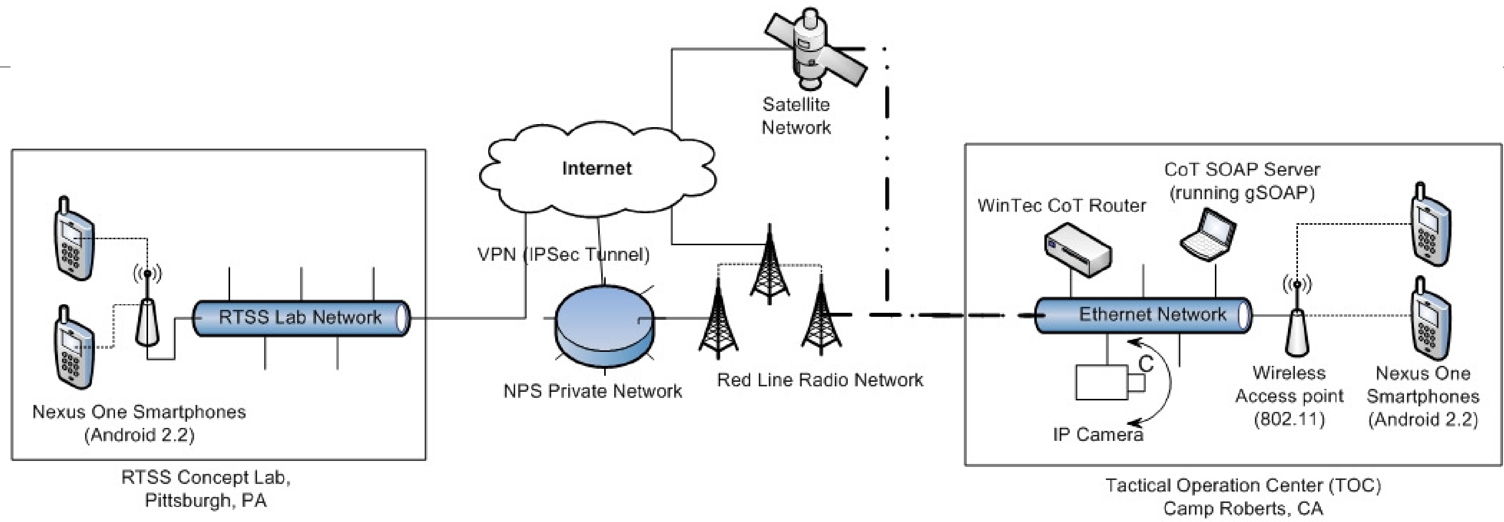
Prototype 2 – CoT SOAP UDP App V2 (Camp Roberts August 2010 TNT) – Mobile Station



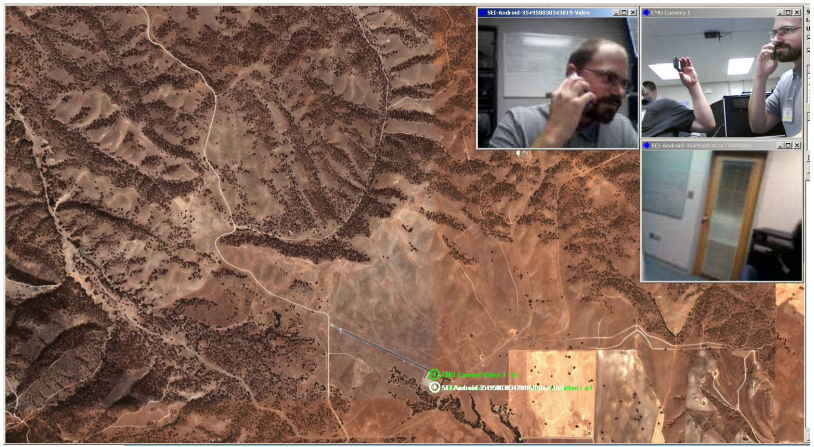
- “Warfighers” disguised as civilian tourists have Android phones that are connected to a Wi-Fi access point (connected to a wave relay via ethernet) on a mobile station (car)
- Phones consume CoT Messages from other assets (UAV)
- Phones also broadcast live video feeds to other phones and the TOC



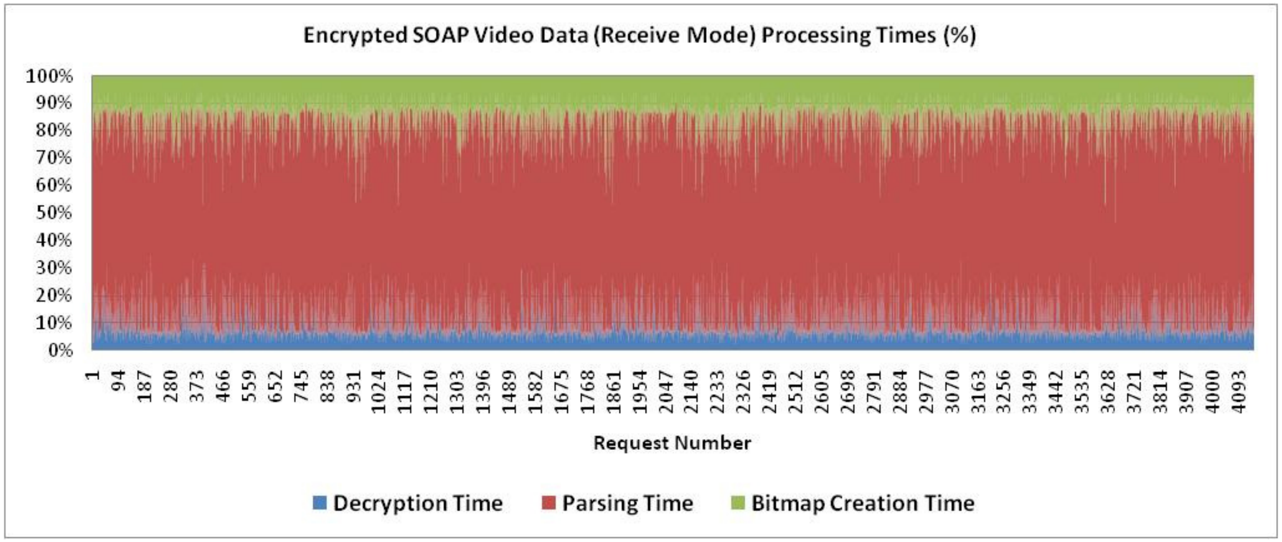
Prototype 3 – CoT SOAP UDP App V2 (Camp Roberts August 2010 TNT) – Two-way Video over VPN



- IP camera and Android phones (inside the TOC, Camp Roberts) capture video frames and convert them into SOAP-over-UDP CoT messages.
- These SOAP messages are sent to both local service consumers (other phones and CoT display) as well as remote service consumers (phones inside the RTSS lab network, Pittsburgh)
- Remote Phones (inside the RTSS lab) can also broadcast live video feeds as SOAP-over-UDP CoT messages to the TOC in Camp Roberts via the RTSS Lab VPN connection

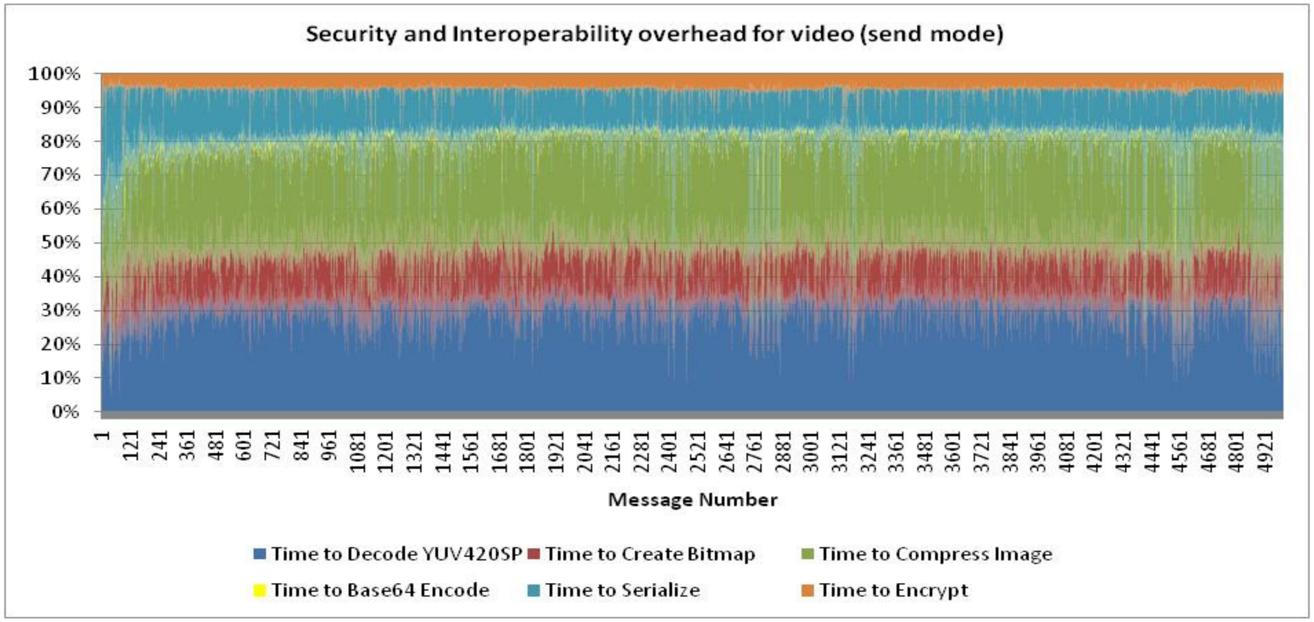


Security and Interoperability overhead for SOAP Video Messages (Receive Mode)



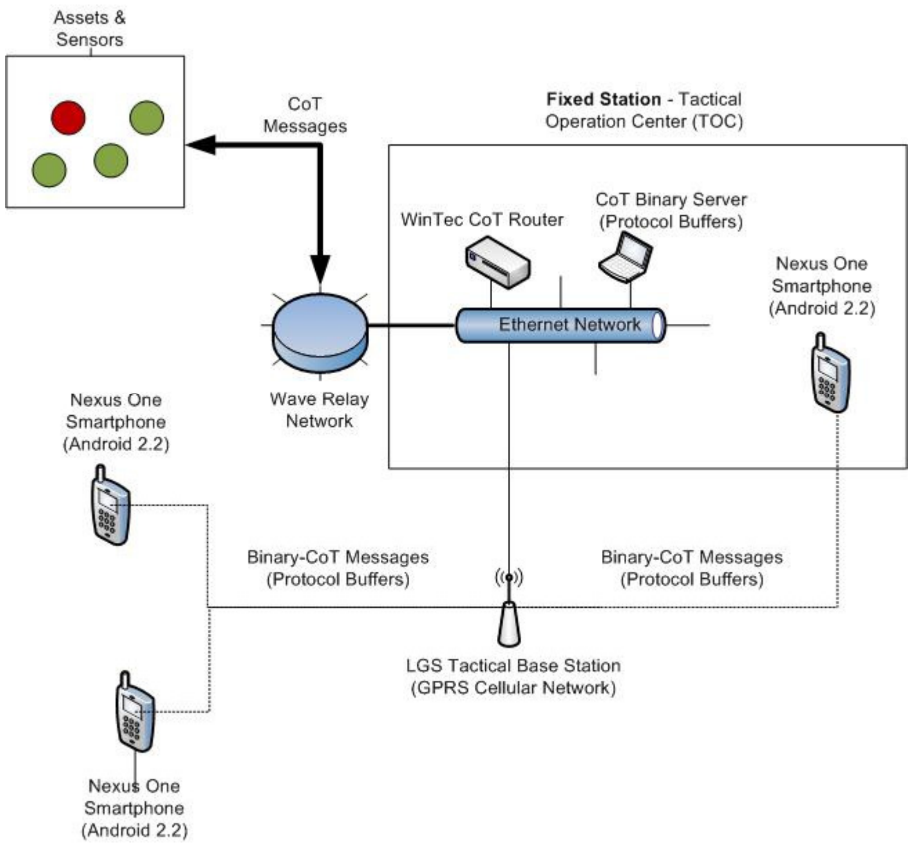
	Average	Std Dev	Median
Decryption Time (ms)	6.074469	4.149628	3.997803
Parsing Time (ms)	52.68427	21.83106	44.12842
Bitmap Creation Time (ms)	10.33385	6.397927	7.141113

Security and Interoperability overhead for SOAP Video Messages (Send Mode)



	Average	Std Dev	Median
Time to Decode UV420SP (ms)	5.485956	1.301711	5.310059
Time to Create Bitmap (ms)	2.659412	0.935272	2.288818
Time to Compress Image (ms)	4.55023	1.458631	5.401611
Time to Base64 Encode (ms)	0.177075	0.785067	0.122071
Time to Serialize (ms)	2.954358	4.221974	2.075196
Time to Encrypt (ms)	0.815698	0.499913	0.701904

Prototype 4 – CoT UDP, Binary Protocol, GPRS (Camp Roberts October 2010 TNT)



- CoT data including imagery sent to and from Android phones are transmitted via an LBS Tactical Base Station using GPRS
- Due to reduced bandwidth, real-time video was not attempted, only still imagery (snapshots) were sent.
- Performance with still images was quite good (approximately 30-40 kbits/sec)

Summary of Observations

Security overhead

- 256 bit AES encryption and decryption at line speed
- Accelerated by use of “native” code
- Significantly less as compared to interoperability overhead (SOAP/XML serialization de-serialization overhead)

Variation in processing time (especially for Java only function such as parsing) is due to

- JVM automatic garbage collection
- Variation in message size for video frames
- Potential problem with kSOAP implementation

IRAD Results – Technical 1

A subset of SOA is practical in tactical environments where bandwidth is high

- Performance of video feeds (using SOAP-based CoT format) is comparable (visually) with the video feed on a standard desktop machine for a threshold of video feeds

To make SOA practical in these situations, we had to make some non-typical SOA choices

- UDP rather than customary TCP to improve video performance
- SOAP vs. REST based web services
  - Currently, REST is built on HTTP/TCP, but there has been some work on REST-UDP
- Use of WS-Security is not easy since implementations are lacking

Moved to binary formats in situations with low bandwidth

IRAD Results – Technical 2

Smartphone (Nexus One) far exceeded our expectations

- Roughly as powerful as a circa 2000 desktop
- Sophisticated software development platform
- Phones with dual core processors are now being developed and released

Our current solution does not address

- Architectural changes to the way CoT data is distributed (that may be need to improve performance)
  - Video frame size (with UDP)
  - UDP is not reliable – not a problem of real-time video but is a problem for text
  - Current phone hardware can be overwhelmed by too many video feeds
  - Not clear if the parsing overhead can be fixed using a better implementation or modifying the existing parser on the phone
- Non-software issues in the field – screen visibility in bright light, radio interference

IRAD Results - Non-technical

May 2010 TNT (Camp Roberts)

- Good feedback from Special Forces personnel
- Requested for more features

August 2010 TNT (Camp Roberts)

- General feeling that we were ahead of others in our development of smartphone capabilities for situational awareness
- By November 2010, we were leapfrogged
- Other consumers of CoT data using our CoT web service

October/November 2010

- Worked with NPS, Naval Special Warfare (NSW) experts to make technology available on proprietary GPRS network
- GPRS sufficient for single images but not for video

Lessons Learned – Mobile Computing

Android is good and open development platform

However, the platform is still evolving

- Performance improvements with new OS versions
- User interface needs to be more responsive
- Lack for hardware-base support for security, but this changing all the time

Architecture must take into account constraints of mobile devices

- The CoT architecture mayneed to be adapted to support constrained devices
  - For example, “on demand” video feeds can reduce bandwidth usage as well as unwanted processing on the phone

Lessons Learned – SOA on Smartphones

Immaturity of SOAP implementations

- kSOAP
  - Lack of support of generated code from WSDL
  - Many key XML features missing or are not directly supported
  - No support for UDP transport, security and other WS standards
- gSOAP
  - Poor architecture for supporting applications that need low-level networking control
  - Problems with UDP support

Reduction in Development time

- Using SOAP can reduce development time by reusing services only if mature implementations are available for the target platform(s)
- Using SOAP can result in unacceptable performance for some class of applications

Current Capabilities of the Android App

Online and cached maps

Asset's geo-location, field of view (FOV) and video feed displayed on a map

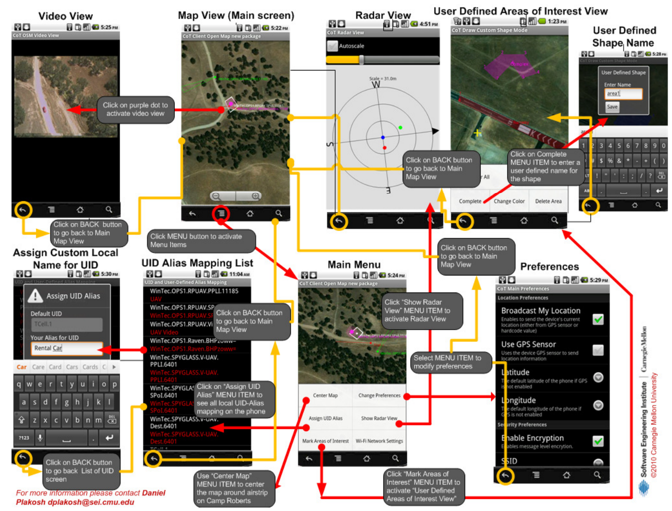
Radar and compass view

User-defined "area of interest" that can be shared in real-time

Real-time video feed from phones

Phone's current geo-location in real-time

256-bit AES encryption



FY11: Warfighter (Edge) Programming of Handheld Devices

Mission Planning

Receive Mission Assignment

Identify Data Sources and Assets Available

Warfighter programming of a mission-specific operational picture

- Problem 1: Information overload
- Problem 2: Information according to warfighter needs and context

Filter and share information as required by mission needs and context

Analyze and report completed mission

Mission Analysis

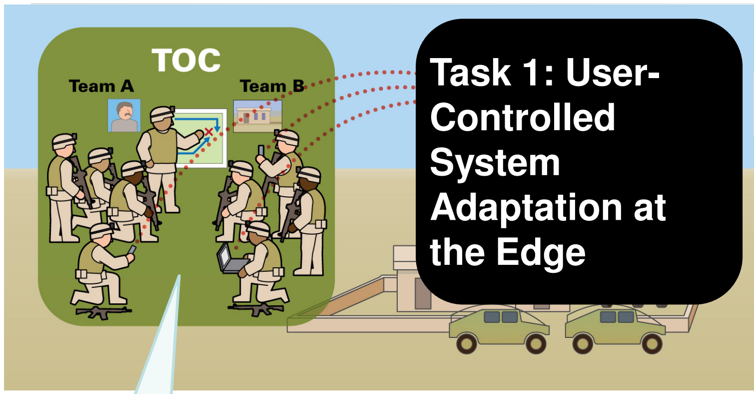
Tailor and use mission-specific operational picture

Justified confidence in warfighter-defined solutions

Mission Execution

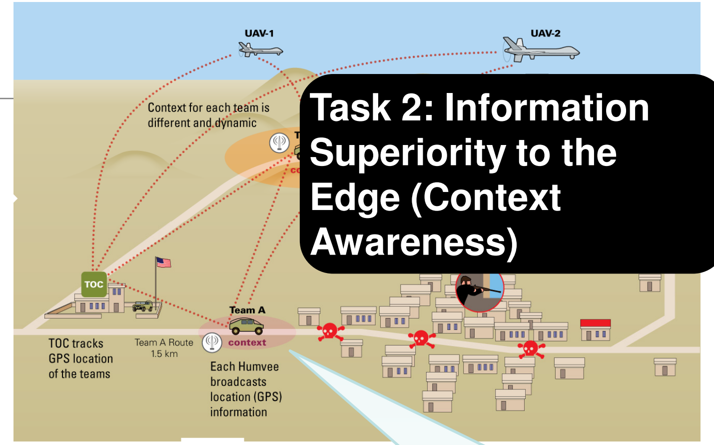
FY12 Planned Work

Mission Planning



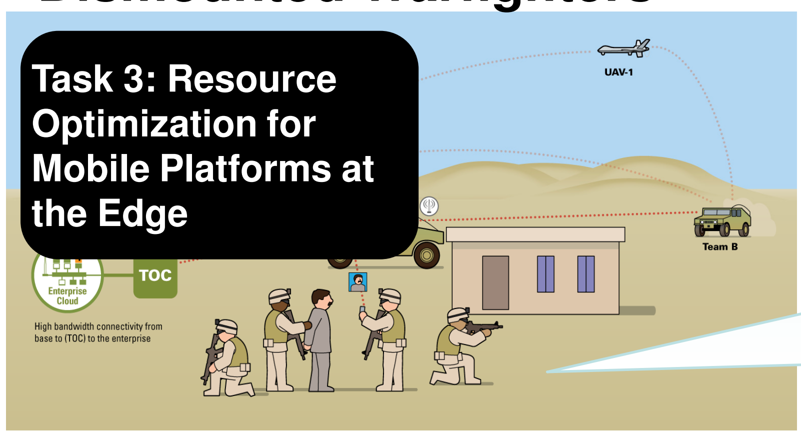
A group of warfighters receives a new mission. Inevitably, the systems they have are not ideally suited to support that mission.

Emerging Situation



Once the warfighters embark on the mission, the circumstances and crucial data for each team changes rapidly.

Dismounted Warfighters



In the field, as warfighters move further away from the enterprise and TOC, computational resources decrease.